Free Space Representation for a Mobile Robot moving on a Rough Terrain

B. Dacre Wright, T.Siméon
LAAS / CNRS
7, Avenue du Colonel Roche
31077 Toulouse - France

Submitted to 93' IEEE ICRA

Abstract

This paper addresses the problem of autonomous navigation for mobile robots on rough terrains [1]. We propose a free space structuring as a promising way to improve the efficiency of path planning for all-terrain mobile robots. This space structuring is based on the characterization of configuration space regions for which the locomotion architecture can guarantee terrain irregularities absorption and stability of the vehicle.

A planner based on this approach has been implemented and experimental results show its interest for practical use.

1 Introduction

There is an increasing interest in developping mobile robots for applications requiring autonomous navigation on natural terrains (planetary exploration [4][5][11], public safety robotics and other hazardous missions [7] [8]). This paper addresses the path planning problem for such all-terrain mobile robots.

Research in motion planing has been very active over the past decade. Most of the work has addressed the problem of finding a collision-free path for manipulators or for mobile robots moving in planar environments (see [6] for an overview). This problem is usually solved by a decomposition of the configuration space into networks of free regions. Graph search techniques and heuristics are then used to search a path from which a safe sequence of motions can be computed.

However, motion planning is almost inexistent in the litterature for mobile robots moving on non-planar terrains. There already exists all-terrain mobile robots which use simple path generation techniques for their autonomous navigation. These approches use the terrain model to characterize the traversability of small terrain patches (roughness of the patch, preferred crossing direction,...). In [4][9], paths are computed from this characterization by gradient propagation techniques. The AMR robot [7] uses a 2D path-planner similar to the one presented in [2] to plan a path that avoids the patches classified as obstacle.

These simple techniques are certainly sufficient for the case of benign terrains. However, they will fail to find a safe path for a robot moving on a rough terrain: in this case, the binary notion of obstacles and non obstacles regions does not hold anymore; the obstacles depend on the ability of the robot to cross over the irregularities of the terrain and their characterization requires to better formalize the constraints acting on the placement of the robot on the terrain.

We are aware of a very few contributions that consider this rather new topic of the path planning problem. A geometric 3D planner is described in [10] for the case of a 3-wheels robot moving on a polygonal terrain. This planner produces paths which not only satisfy collision-free constraints, but also stability and non-holonomic constraints. This planning technique is derived from the approach described in [1] for the planar case; it consists of exploring a discretized representation of the **C**onfiguration **S**pace to incrementally search for a safe trajectory. A similar formulation of the problem is reported in [8] for the Intelligent Locomotion of a four tiltable track robot. As mentionned by the authors, this incremental search of a safe trajectory can be very time consuming and we feel that having a preprocessing step aimed at discovering the "good" **CS**-regions could be used to better guide a more complex path-planning method.

The main contribution of this paper is to produce such a preprocessing. We provide a characterization

[1] This work has been done in the framework of the Automatic Planetary Rover project conducted by the French Spatial Agency. It was partially supported by C.N.E.S. and by the ESPRIT Project PROMotion 6546.

of the **CS**-regions for which the locomotion architecture of the robot can guarantee terrain irregularities absorption and stability of the vehicle.

The paper is organized as follows: After a presentation of the robot and terrain models, Section 2 gives a more formal statement of the placement problem and of the constraints acting on the placements of the robot. Section 3 presents an algorithm that characterizes the admissibility of a given **CS**-region. Finally, Section 4 combines these results to obtain an octree representation of these admissible regions. The algorithm have been implemented and experimental results are presented at the end of the section.

2 Formalization of the placement problem.

2.1 General statement.

We consider an articulated robot consisting of a body and n wheels attached to it by suspensions. These suspensions allow the wheels to keep a better contact to the terrain, but on the other hand, the placement of the robot becomes complex : it results from the interaction with the terrain and the balance of the suspensions, which are modeled by springs.

The terrain is known by a discrete elevation map, that is the elevation values z on a discrete regular grid in (x, y).

In the sequel, we consider that 3 conditions are required for a placement to be valid :

(**C1**) all the wheels remain in contact with the ground.

(**C2**) the springs cannot be stretched beyond some limit length.

(**C3**) the robot does not tip-over, which requires that the projection of its gravity center remains inside the convex hull of the projections of all the contact points (called the support polygon).

Our purpose is to determine domains of the configuration space which verify these conditions.

2.2 The placement parameters.

Let us define a local frame $(G, \mathbf{u}, \mathbf{v}, \mathbf{w})$ attached to the robot, where G is the gravity center of the robot, and $(\mathbf{u}, \mathbf{v}, \mathbf{w})$ are respectively its longitudinal, lateral and vertical axis (see Fig. 1).

We assume that, when all the springs are in their steady state, the wheels belong to the same plane $\mathcal{P}_S$ perpendicular to w axis, at height w_0 in the local frame. Then, the coordinates of wheel i are $(u_i, v_i, w_0 + l_i)$ where u_i and v_i are some fixed values related to the geometry of the robot, and l_i is the algebraic extension of the spring ($l_i = 0$ being its natural length). That means that the springs always keep a vertical orientation in the robot's frame. Additionally, we assume that the positions (u_i, v_i) of the wheels are symmetrically distributed around the vehicle. Fig. 1 shows an example of a 6-wheels robot, with the parameters detailed above.

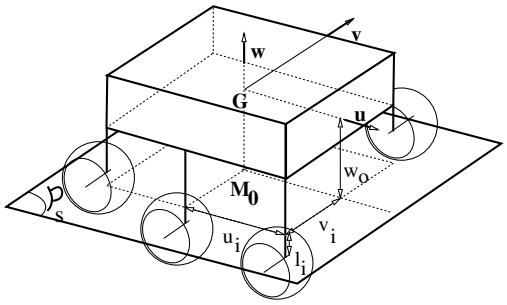


Figure 1: Model of the vehicle

A position of the robot's body is defined by the 6-dimensional vector $\mathbf{p} = (x_G, y_G, z_G, \theta, \phi, \psi)$, where (x_G, y_G, z_G) are the coordinates of point G, and (θ, ϕ, ψ) are respectively the horizontal orientation of u axis, the roll angle and the pitch angle[2]. These parameters define the transform matrix from the robot frame to the space frame $(O, \mathbf{x}, \mathbf{y}, \mathbf{z})$:

$$M(\mathbf{p}) = T(x_G, y_G, z_G) R(\mathbf{w}, \theta) R(\mathbf{v}, \psi) R(\mathbf{u}, \phi).$$

Therefore, a complete placement of the robot is given by $(6 + n)$ independant parameters :

- the 6-dimensional vector $\mathbf{p} = (x_G, y_G, z_G, \theta, \phi, \psi)$ defining the position of the robot's body in the space,
- and the n spring extensions l_i, giving the wheel positions.

Nevertheless, the interactions with the terrain constrain these parameters and therefore reduce the dimension of the robot's configuration space. This will be developped in the next section.

2.3 Definition of the configuration space.

The placement of the robot results from its weight and the reaction of the ground exerted through the springs.

The static equilibrium state is reached when the total energy of the robot is minimized. In the following, we consider a simple energy function, including only

[2] $|\phi| < \frac{\pi}{2}$, $|\psi| < \frac{\pi}{2}$.

the compression energy of the springs:

$$\mathcal{E}((l_i)_{i=1\ldots6}) = \sum_{i=1}^{6} k{l_i}^2. \quad (1)$$

where k is the stiffness coefficient of the springs.

Moreover, the wheels are supposed to stay over the terrain, which constrains their positions. Let $z = f(x,y)$ be an analytic expression of the terrain, the free space being the set of points such that $z > f(x,y)$. The position of wheel i is defined by the position $\mathbf{p}$ of the robot's body and the length l_i of its spring. We call $W_i(p, l_i)$ the volume occupied by the wheel. The placement of wheel i is strictly expressed as :

$$\forall M(x,y,z) \in W_i(\mathbf{p}, l_i), \quad z \geq f(x,y).$$

In order to keep a simple constraint expression, we approximate the wheel's shape by a point[3] denoted $M_i(x_i, y_i, z_i)$. Hence the constraint is only applied to this contact point M_i. Moreover, we are only interested in the placements which keep all the wheels in contact with the terrain (condition **C1**). Therefore, the contact constraints become :

$$z_i = f(x_i, y_i) \quad (2)$$

where x_i,y_i and z_i are related to the placement $\mathbf{p}$ and to the length l_i by:

$$(x_i, y_i, z_i)^T = M(\mathbf{p})(u_i, v_i, w_0 + l_i)^T$$

For given values of the three parameters (x_G, y_G, θ), the remaining parameters $(z_G, \phi, \psi, (l_i)_{i=1\ldots6})$ result from the minimization of energy (1) under equality constraints (2). Thus we consider the three dimensional configuration space induced by the parameters (x_G, y_G, θ). Any configuration defines the placement of the robot as follows :

Definition 1 *For any configuration* $\mathbf{q} = (x_G, y_G, \theta) \in CS = \mathbf{R}^2 \times S^1$, *the associated placement of the robot is the set of parameters* $(\mathbf{p}, (l_i)_{i=1\ldots n})$ *such that :*

- $\mathcal{E}((l_i)_{i=1\ldots n})$ *is minimal.*
- $\forall i \in [1,n], z_i = f(x_i, y_i)$.

2.4 The admissible configurations.

There remains to link together the requirements **C2** and **C3** defined in section 2.1 and the configuration parameters (x_G, y_G, θ).

[3]or by a sphere, with an appropriate change of function f, which corresponds to growing the terrain by the radius of the wheels.

Constraint **C2** limits the n spring elongations l_i to a given value denoted L_{max}. According to constraint **C3**, the robot is stable when the vertical projection of its gravity center belongs to the support polygon. The shape of this support polygon is clearly a function of ϕ, ψ and the lengths $(l_i)_{i=1\ldots n}$. However, we assume that the less stable position is obtained when all the springs are the longest[4]. The stability thus only depends on ϕ and ψ, and we can determine a subset $S \subset]-\frac{\pi}{2}, \frac{\pi}{2}[^2$ such that for any pair $(\phi, \psi) \in S$, the robot position is stable.

We can now define precisely the admissible space of the robot.

Definition 2 *The admissible subset* ACS *is the set of configurations* $\mathbf{q} \in CS$ *such that :*

- $\forall i \in [1,n], |l_i| \leq L_{max}$.
- $(\phi, \psi) \in S$.

In the remaining, we propose a method to built a hierarchical decomposition of ACS relying on the analysis of the admissibility of a domain $\Delta x \times \Delta y \times \Delta\theta$ in CS.

3 Admissiblity of a CS region.

We consider now a region $R_{CS} = [x_1, x_2] \times [y_1, y_2] \times [\theta_1, \theta_2] \subset CS$. Our purpose is to determine if the region R_{CS} is in ACS or not.

3.1 Overview.

Similarly to the configuration vector $\mathbf{q}$, we define the vector $\mathbf{r}$ of the remaining placement parameters : $\mathbf{r} = (z_G, \phi, \psi, (l_i)_{i=1\ldots n})$.

Let Δr_0 be a domain which is guaranteed to contain any admissible placement. For example, l_i necessarily belongs to $\Delta l_i = [-L_{max}, L_{max}]$.

For any wheel i, we can determine the area in the 3-dimensional natural space of all its positions for all placements in $R_{CS} \times \Delta\mathbf{r}_0$. The vertical projection of this area gives an (x,y)-region called contact region $\mathcal{R}_i$. The contact of the wheel onto the terrain belongs to the set of points (x,y,z) of the terrain such that $(x,y) \in \mathcal{R}_i$. The elevation map is used to compute a sort of bounding box of this terrain portion, called contact envelope $\mathcal{V}_i$ (cf Section 3.2).

The definition of a placement given in Section 2.3 establishes relations between the placement parameters and the elevation of the terrain. These relations allow to deduce a new placement interval $\Delta\mathbf{r}$ from the contact envelopes (cf Section 3.3). The thicker the envelopes are, the larger are the parameter intervals $\Delta\mathbf{r}$.

[4]This hypothesis is not restrictive and facilitates the computation.

For this reason, the contact envelopes must be as close as possible to the terrain.

The obtained placement domain $R_{CS} \times \Delta\mathbf{r}$ necessarily contains all the possible placements of the robot in region R_{CS}, whatever the terrain's shape inside the contact envelopes. If this interval is in accordance with the constraints **C2** and **C3**, R_{CS} is guaranteed to be in ACS. Otherwise, we consider that the region R_{CS} could contain placements which would not verify the constraints, and we declare R_{CS} to be forbidden.

3.2 The contact envelopes.

In this section, we show how to compute, for each wheel, the terrain envelope where the contact should occur. We first determine an average plane $\mathcal{P}_a$ from the points of the terrain over all the contact regions. Then we compute for each wheel the limit values of the terrain elevation relatively to plane $\mathcal{P}_a$.

3.2.1 The contact regions $(\mathcal{R}_i)_{i=1\ldots n}$.

We first consider the influence of ϕ, ψ and $(l_i)_{i=1\ldots n}$. Let $(G', \mathbf{u}', \mathbf{v}')$ be the horizontal frame obtained from the vertical projection of $(G, \mathbf{u}, \mathbf{v})$,(see Fig. 2). In this frame, the vertical projection of wheel i is given by the two first coordinates of $R(\mathbf{v}, \psi)R(\mathbf{u}, \phi)(u_i, v_i, w_0 + l_i)$. When ϕ, ψ and l_i vary in $\Delta\phi$, $\Delta\psi$ and Δl_i, this projection remains in an area which we approximate by the minimum enclosing rectangle $[u'_1, u'_2] \times [v'_1, v'_2]$.

When $\mathbf{q}$ varies in R_{CS}, this rectangle sweeps an area in accordance with the values of Δx, Δy and $\Delta\theta$ (see Fig. 2). This area is approximated by a polygon which defines the expected contact region $\mathcal{R}_i$.

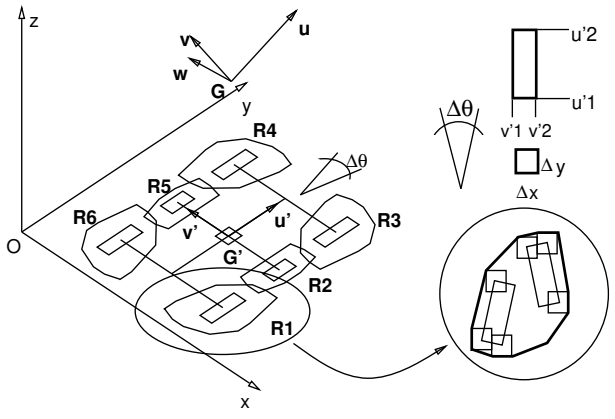


Figure 2: Computation of the contact regions

3.2.2 The average plane $\mathcal{P}_a$.

Now, we compute a plane approximation over the union of all contact regions. For this purpose, we collect the points of the elevation map over all the contact regions. The approximation plane is computed by a least square method. Let A, B and X be matrices defined as follows :

$$A = \begin{pmatrix} x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots \\ x_p & y_p & 1 \end{pmatrix}, B = \begin{pmatrix} z_1 \\ \vdots \\ z_p \end{pmatrix}, X = \begin{pmatrix} a \\ b \\ c \end{pmatrix}$$

where $((x_i, y_i, z_i))_{i=1\ldots p}$ are the collected points of the elevation map, and a, b and c are the coefficients of the plane equation : $\mathbf{z = ax + by + c}$. Vector X is given by the matrix equation :

$$AX = B.$$

whose solution is :

$$X = (A^T A)^{-1} A^T B.$$

with

$$\begin{aligned} A^T A &= \begin{pmatrix} \sum x_i^2 & \sum x_i y_i & \sum x_i \\ \sum x_i y_i & \sum y_i^2 & \sum y_i \\ \sum x_i & \sum y_i & p \end{pmatrix} \\ A^T B &= \begin{pmatrix} \sum x_i z_i \\ \sum y_i z_i \\ \sum z_i \end{pmatrix} \end{aligned}$$

Let us just mention that, in order to spare memory, we compute directly the matrices $A^T A$ and $A^T B$, by adding for each new point, the associated terms to the matrices. When all the points have been collected, there only remains to inverse a 3×3 symetrical matrix, and to multiply it by a 3-dimensional vector.

The coefficients a, b, c define the plane which minimizes the quadratic mean of the vertical distances to the elevation map data. This average plane denoted $\mathcal{P}_a$ provides a placement called $\mathbf{p}_a$:

$$\mathbf{p_a} = (x_a, y_a, z_a, \theta_a, \phi_a, \psi_a)$$

with:

$$\begin{cases} x_a &= \frac{x_1 + x_2}{2} \\ y_a &= \frac{y_1 + y_2}{2} \\ z_a &= a x_a + b x_a + c \\ \theta_a &= \frac{\theta_1 + \theta_2}{2} \\ \phi_a &= \arctan((b \cos\theta_a - a \sin\theta_a) \cos\psi_a) \\ \psi_a &= -\arctan(a \cos\theta_a + b \sin\theta_a) \end{cases}$$

This placement is not supposed neither to have any optimal property nor to correspond to the mean values

on each placement parameters. It is only a good indication on what could be the placement of the robot upon the contact regions. $\mathbf{p}_a$ will be used as a reference position to determine the bounds of the placement parameters around it.

3.2.3 The contact envelopes $(\mathcal{V}_i)_{i=1\ldots n}$.

For each contact region $\mathcal{R}_i$, we compute the extreme elevation differences between the points of the elevation map and $\mathcal{P}_a$. For any point $M(x,y,z)$ in the elevation map, this difference is $\delta z = z-(ax+by+c)$. Let δz_i^{min} and δz_i^{max} be the extreme values of δz on the contact region $\mathcal{R}_i$. The contact envelope is :

$$\mathcal{V}_i = \left\{ M \in \mathbf{R}^3 \;\middle/\; \begin{array}{l} (x,y) \in \mathcal{R}_i \quad \text{and} \\ z-(ax+by+c) \in [\delta z_i^{min}, \delta z_i^{max}] \end{array} \right\}$$

Fig. 3 illustrates this notion of contact envelopes. Notice that all the envelopes have the same orientation, defined by the common average plane $\mathcal{P}_a$, but each has a specific depth, since it is computed separatly on each contact region. However, one may wonder why a different mean plane is not computed on each region, which would provide envelopes even closer to the terrain. The reason is that in order to be valid on the whole region R_{CS}, the computed domain $\mathbf{\Delta r}$ must be independant of (x,y,θ). This will be clarified in the following section.

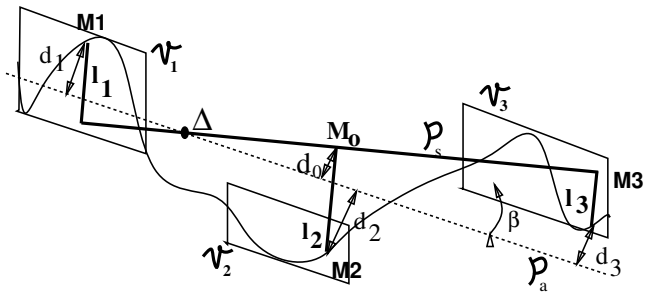


Figure 3: Placement variations around the average plane

3.3 The placement domain Δr.

We now use equation (1) which characterizes a placement, to give some bounds on each parameter, according to the dimensions of the contact envelopes.

According to Definition 1, the placement minimizes the robot's energy $\mathcal{E}$. The partial derivates of $\mathcal{E}$ with respect to z_G, ϕ and ψ need therefore to be null. As these differentiations of $\mathcal{E}$ yield too complex expressions, we introduce three new parameters to define the relative placement of the robot with respect to $\mathcal{P}_a$ (Section 3.3.1). The bounds on these parameters can be easily obtained (Section 3.3.2), and then related to the bounds of $\mathbf{\Delta r}$ (Section 3.3.3).

3.3.1 Relative placement from the average plane.

We introduce three parameters (α, β, d_0) characterizing the position of the steady state plane $\mathcal{P}_S$ of the robot (whose equation is $w = w_0$ in the robot's frame) relatively to the average plane $\mathcal{P}_a$. Let Δ be the line $\mathcal{P}_S \cap \mathcal{P}_a$ and β the angle between the two planes. We denote by α the orientation of the robot in plane $\mathcal{P}_S$ relatively to Δ. Finally, we call d_0 the distance of point $M_0(0,0,w_0)$ in the robot's frame to plane $\mathcal{P}_a$.

Actually, we are interested in the limits of parameters d_0 and β (which respectively correspond to the vertical position and the attitude of the robot) for any value of α.

Fig. 3 illustrates these notations, and will give a support to the following developments. Three contact points M_1, M_2 and M_3 are represented, with the associated distances to $\mathcal{P}_a$ and the spring lengths l_i, which are the distances to $\mathcal{P}_S$.

3.3.2 Determination of the bounds on the robot's position.

Let $(d_i)_{i=1\ldots n}$ be the distances of the wheels to $\mathcal{P}_a$: $d_i = \cos\phi_a \cos\psi_a \delta z_i$. The length of each spring can be related to the associated d_i by:

$$l_i = \frac{1}{\cos\beta}(d_i - d_0) - \tan\beta(\sin\alpha u_i + \cos\alpha v_i). \quad (3)$$

The differentiation of $\mathcal{E}$ with respect to d_0 yields :

$$\sum_{i=1}^{n}(d_i - d_0) - \sin\beta \sum_{i=1}^{n}(\sin\alpha u_i + \cos\alpha v_i) = 0$$

Let us remind that the wheels are assumed to be symetrically placed around the vehicle. This implies : $\sum u_i = \sum v_i = \sum u_i v_i = 0$, and we finally obtain :

$$d_0 = \frac{1}{n}\sum_{i=1}^{n} d_i.$$

Hence d_0 is between the mean of minimal values of d_i on each contact envelope, and the mean of the maximal values. Since d_i can be bounded according to the elevation bounds $[\delta z_i^{min}, \delta z_i^{max}]$ computed in Section 3.2.3, the extreme values of d_0 are:

$$d_0^{min} = \frac{1}{n}\cos\phi_a \cos\psi_a \sum_{i=1}^{n} \delta z_i^{min}$$

$$d_0^{max} = \frac{1}{n}\cos\phi_a\cos\psi_a\sum_{i=1}^{n}\delta z_i^{max}$$

Similarly, the differentiation by β leads to :

$$f(\beta) = \frac{\tan\beta}{\cos\beta} = \frac{B}{C-A}$$

where
$$\begin{cases} A = \sum_{i=1}^{n}(d_i - d_0)^2 \\ B = \sin\alpha\sum_{i=1}^{n} d_i u_i + \cos\alpha\sum_{i=1}^{n} d_i v_i \\ C = \sin^2\alpha\sum_{i=1}^{n} u_i^2 + \cos^2\alpha\sum_{i=1}^{n} v_i^2 \end{cases}$$

As function f is monotonous on $]-\frac{\pi}{2},\frac{\pi}{2}[$, we can deduce the maximal angle β^{max} between $\mathcal{P}_S$ and $\mathcal{P}_a$ from the maximal value of $\frac{B}{C-A}$.

Let us first give some bounds to A, B and C, for any value of α :[5]

$$\max|A| = \max(\sum_{i=1}^{n}(d_i^{max} - d_0^{min})^2, \sum_{i=1}^{n}(d_0^{max} - d_i^{min})^2)$$

$$\max|B| = \sqrt{\max(\sum_{i=1}^{n} d_i u_i)^2 + \max(\sum_{i=1}^{n} d_i v_i)^2}$$

$$\min|C| = \min\sum_{i=1}^{n} u_i^2, \sum_{i=1}^{n} v_i^2$$

All these values may be computed from the bounds on $(d_i)_{i=1\ldots n}$[6].

The first condition to be sure there exists a bound to β is to insure that $\max|A| < \min|C|$ (otherwise there could exist some terrain configuration for which $\beta = \pm\frac{\pi}{2}$). Under this condition, the bounds on β can be expressed as follows :

$$\beta^{max} = 2\arctan[-\frac{1}{K}(1-\sqrt{K^2+1})]$$

$$\text{with} \quad K = \frac{\max|B|}{\min|C| - \max|A|}$$

3.3.3 The new placement domain Δr.

The bounds on ϕ and ψ can be related to β_{max}. It can be easily shown that : $\cos\beta = \cos(\phi-\phi_a)\cos(\psi-\psi_a)$. Hence $\max|\phi-\phi_a| = \max|\psi-\psi_a| = \beta_{max}$ which gives :

$$\Delta\phi = [\phi_a - \beta^{max}, \phi_a + \beta^{max}]$$
$$\Delta\psi = [\psi_a - \beta^{max}, \psi_a + \beta^{max}]$$

[5] In order to keep simple expressions, we give them according to the bounds on d_i, since $d_i^{min} = \cos\phi_a\cos\psi_a\delta z_i^{min}$ and $d_i^{max} = \cos\phi_a\cos\psi_a\delta z_i^{max}$.

[6] In the expression of $max|B|$, the maximal value of $\sum_{i=1}^{n} d_i u_i$ is obtained by adding $d_i^{max}u_i$ when $u_i > 0$, $d_i^{min}u_i$ otherwise. The minimal value is obtained with $d_i^{min}u_i$ when $u_i > 0$ and $d_i^{max}u_i$ otherwise. The same reasoning can be applied to $\sum_{i=1}^{n} d_i v_i$.

Finally, the bounds on $(l_i)_{i=1\ldots n}$ are obtained by minimizing and maximizing expression (3) :

$$\forall i, \quad \Delta l_i = [l_i^{min}, l_i^{max}]$$

with:

$$\begin{cases} l_i^{min} = \frac{1}{\cos\beta^{max}}(d_i^{min} - d_0^{max}) - \tan\beta^{max}\sqrt{u_i^2+v_i^2} \\ l_i^{max} = \frac{1}{\cos\beta^{max}}(d_i^{max} - d_0^{min}) + \tan\beta^{max}\sqrt{u_i^2+v_i^2} \end{cases}$$

The admissibility of R_{CS} is now very simple to establish, according to Definition 2. Region R_{CS} is declared admissible if :

$$\forall i, \quad \Delta l_i \subset [-L_{max}, L_{max}] \qquad (4)$$
$$\Delta\phi \times \Delta\psi \subset S \qquad (5)$$

Proposition 1 *For any cubic region $R_{CS} \subset CS$, if the computed placement domain Δr verifies conditions (4) and (5), then $R_{CS} \subset ACS$.*

4 Hierarchical decomposition of CS.

The analysis of CS-regions is now used to compute an approximate cell decomposition of ACS. This cell decomposition relies on an octree decomposition of CS.

4.1 The octree structure.

Octree-type representations have already been used for path-planning; see for example [3]. An octree is a recursive decomposition of a cubic space into subcubes. It is a tree of degree eight whose nodes represent in our case cubic regions of CS, also called voxels.

Any voxel can be analyzed according to Proposition 1. If the voxel is in ACS, we declare it FREE. Otherwise, we cannot conclude, and the voxel is declared MIXED[7]. It must be subdivided into eight smaller voxels, for which the test will eventually be successful. This recursive decomposition is performed up to a level p_{max} for which the approximation is fine enought.

The approximate decomposition of ACS consists of all the FREE voxels.

4.2 The construction methods.

There mainly exist two methods for building an octree :

The bottom-up method : the voxels of level p_{max} are first analyzed. The octree is then recursively deduced from the finest level : any voxel of level i is declared FREE if its eight sons of level $i+1$ are FREE. Otherwise, it is declared MIXED. This method has a major drawback : it requires to test

[7] Note that our octree does not contain FULL voxels.

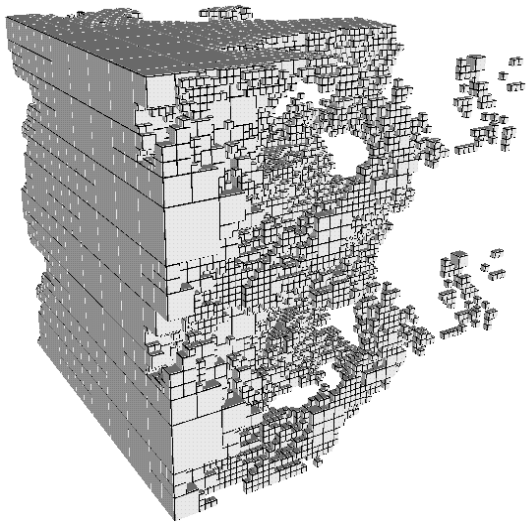


Figure 4: Octree decomposition of ACS associated to the environment of Figure 5.

a great amount of voxels at the finest level, even in areas where such a fine decomposition is not needed.

The top-down method : the root node (level 0) is first analyzed and is recursively decomposed up to level p_{max}. If a voxel of level i is found to be FREE, it is no longer decomposed. Otherwise it is considered as MIXED and must be subdivided into eight voxels of level $i+1$ which are analyzed in turn. However, in our case, large CS-regions are unlikely to be in ACS and their analysis would require to consider an important amount of elevation map datas.

We choose to combine both methods, in order to avoid their respective drawbacks. A reference level, called p_0, is choosen in the octree. We first analyze all the voxels of level p_0, and we recursively refine it in the top-down fashion. Then the levels $i < p_0$ are built according to the bottom-up method.

The choice of a good reference level leads to a minimum number of useless test operations. This reference level mostly depends on the complexity of the terrain. Nevertheless the level where voxels have a size of the order of the robot's length gives usually the best results.

However, searching for a free path between 2 configurations does not require to compute a complete free space representation. We propose in the next section a search method relying on an incremental building of the octree.

4.3 Incremental search for an admissible path.

Let q_s and q_g be respectively the start and goal configurations. We first create in the octree the largest FREE voxels V_s and V_g containing q_s and q_g. Path search from V_s to V_g is performed by a classical A^* algorithm.

At each step of the search, the FREE neighbours of the current voxel are expanded as follows : Let V be the current voxel, belonging to level i, and let V' be one of its neighbours at the same level. If V' is FREE, we recursively analyze its father in order to determine the largest FREE voxel adjacent to V. In the other case, we analyze, among its sons, the 4 voxels adjacent to V. The free sons are appended to the list of neighbours. The MIXED ones are recursively analyzed up to level p_{max}. This procedure is applied to the 6 neighbours of V, and returns the complete list of the largest FREE voxels adjacent to V.

Finally, the search returns a sequence of large voxels connecting V_s to V_g.

4.4 Experimental results.

Figures 4 shows the octree computed for the environment of Figure 5 with $p_{max} = 6$. The terrain is a 64×64 elevation map.

The path search technique described above was used to improve the efficiency of a path planner based on an extension of [10]. Figure 5 shows a trajectory obtained for a 6-wheeled robot. The total time for planning this trajectory is about 1 minute on a Silicon Graphics INDIGO workstation.

5 Conclusion.

In this paper, we have addressed a rather new topic of the path-planning problem, ie. computing a safe path for a mobile robot moving on a rough terrain. We proposed a method, using configuration space, to characterize the placements for which the locomotion architecture of the robot can guarantee terrain irregularities absorption and stability of the vehicle.

We believe that the free space representation proposed in this work captures the essential constraints acting on the placement of the robot over the terrain. Therefore, the CS-regions resulting from the path search can be used to limit the search space for more sophisticated (but also more computationally expensive) path planners which consider additional constraints such as collision or non-holonomic constraints.

Natural extensions of this work would deal with a better integration of these different planning techniques to obtain a more efficient path planner.

References

[1] J. Barraquand and J.C. Latombe. On non-holonomic mobile robots and optimal maneuvering. *Revue d'Intelligence Artificielle*, 3(2), 1989.

[2] J. Barraquand and J.C. Latombe. Robot Motion Planning: A distributed Representation Approach. *The International Journal of Robotics Research*, 10(5), 1991.

[3] B. Faverjon. Object Level programming of industrial robots. In *IEEE International Conference on Robotics and Automation, San Francisco (USA)*, 1986.

[4] E. Gat, M. Slack, D.P. Miller and R.J. Firby. Path planning and execution monitoring for a Planetary rover. In *IEEE International Conference on Robotics and Automation, Cincinnati (USA)*, 1990.

[5] G. Giralt and L. Boissier. The French Planetary Rover VAP: Concept and Current Developments. In *IEEE International Conference on Intelligent Robots and Systems, Raleigh (USA)*, July 92.

[6] J.C Latombe. *Robot Motion Planning.* Kluwer Pub., 1990.

[7] R. Laurette, A. de St. Vincent, R. Alami, R. Chatila and V. Perebaskine. Supervision and Control of the AMR intervention robot. In *Fifth International Conference on Advanced Robotics, Pisa (Italy)*, 1991

[8] M. Iagolnitzer, F. Richard, J.F. Samson and P. Tournassoud Locomotion of an all terrain mobile robot. In *IEEE International Conference on Robotics and Automation, Nice (France)*, 1992.

[9] E. Schalit. ARCANE: Towards utonomous Navigation on rough terrains In *IEEE International Conference on Robotics and Automation, Nice (France)*, 1992.

[10] T. Siméon. Motion planning for a non holonomic mobile robot on three dimensional terrains. In *IEEE International Workshop on Intelligent Robots and Systems, Osaka (Japan)*, November 91.

[11] R. Simmons and E. Krotkov An integrated walking system for the Ambler Planetary Rover. In *IEEE International Conference on Robotics and Automation, Sacramento (USA)*, 1991.

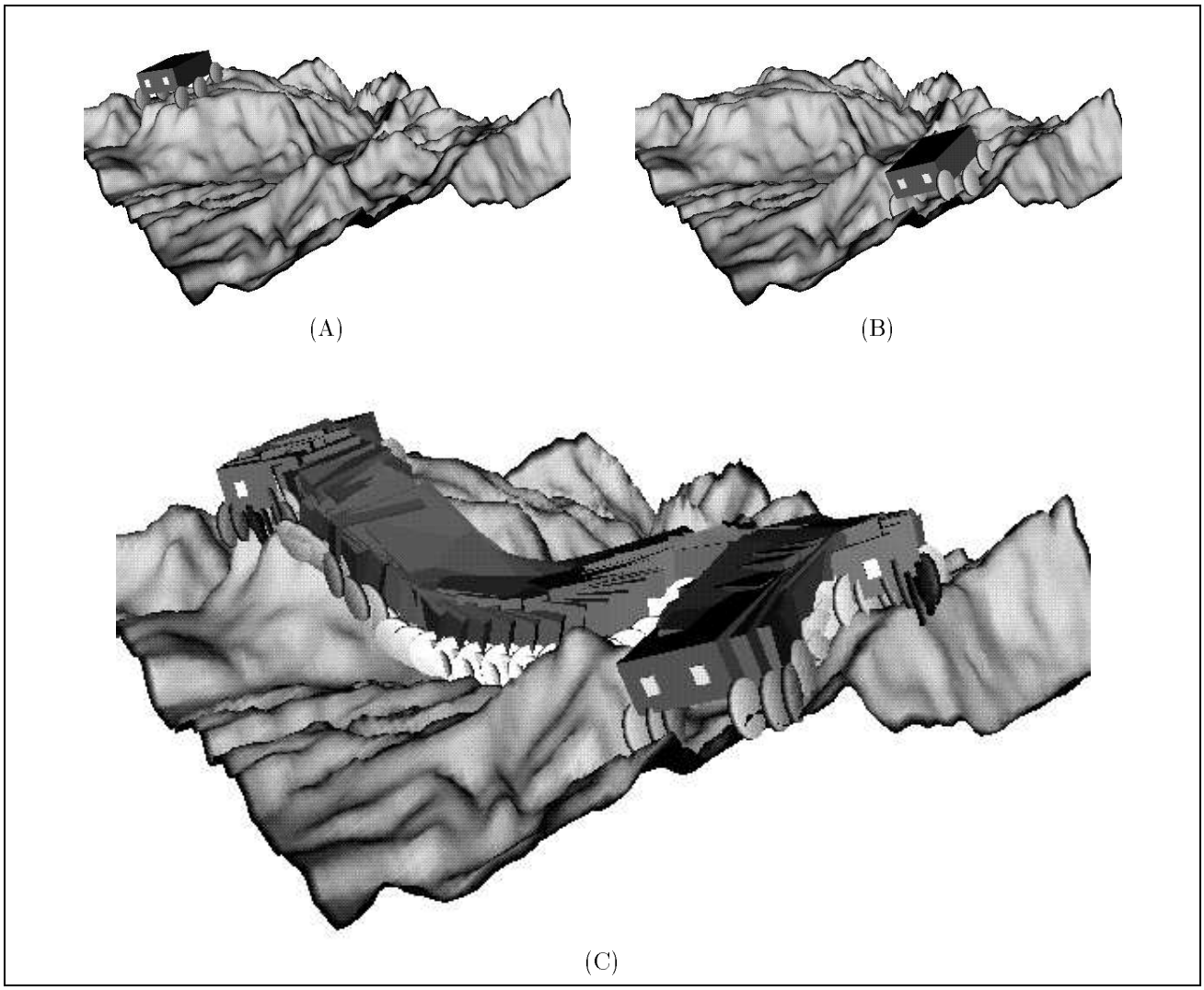


Figure 5: (A) The initial consiguration. (B) The goal configuration. (C) The trajectory obtained for a six wheels robot by using the free-space structuring described here.